2014-12

A comparison of techniques for optimal infrastructure restoration

Schulze, Carsten

Monterey, California: Naval Postgraduate School

http://hdl.handle.net/10945/44665

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A COMPARISON OF TECHNIQUES FOR OPTIMAL INFRASTRUCTURE RESTORATION

by

Carsten Schulze

December 2014

Thesis Co-Advisors: David L. Alderson
Nedialko B. Dimitrov

Second Reader: Michael Atkinson

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704–0188*

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE 12-19-2014	3. REPORT TYPE AND DATES COVERED Master's Thesis 05-01-2014 to 12-19-2014

4. TITLE AND SUBTITLE A COMPARISON OF TECHNIQUES FOR OPTIMAL INFRASTRUCTURE RESTORATION	5. FUNDING NUMBERS
6. AUTHOR(S) Carsten Schulze	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Major disruptions such as terror attacks, natural disasters and human failures can have large impacts on critical infrastructure. The rapid reconstitution of those infrastructure systems after a major disruption is crucial to minimize the impact of the disaster. This thesis compares two different modeling techniques to minimize the cost for reconstructing the infrastructure system. The first technique uses a mixed integer linear program to minimize the operation cost of a infrastructure system. The second technique is a graph-based approach in which the vertices of a meta graph represent different operating states for the infrastructure system, and edges between vertices represent possible transitions between states (e.g., the repair of one or more infrastructure components). In this context, optimal restoration of the infrastructure system corresponds to finding the best (e.g., minimum cost) path from an initial damaged state to a fully restored state. We consider two different ways of finding the shortest path in this meta graph, specifically Dijkstra's algorithm and the A-star algorithm. We compare these techniques in terms of quality of solution and required computation time.

14. SUBJECT TERMS infrastructure, restoration, optimization	15. NUMBER OF PAGES 65
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU

THIS PAGE INTENTIONALLY LEFT BLANK

A COMPARISON OF TECHNIQUES FOR OPTIMAL INFRASTRUCTURE RESTORATION

Carsten Schulze
Major, German Army
Diplom - Ingenieur, Bundeswehr University Hamburg, 2005

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
December 2014

Author:	Carsten Schulze
Approved by:	David L. Alderson, Ph.D. Thesis Co-Advisor
	Nedialko B. Dimitrov, Ph.D. Thesis Co-Advisor
	Michael Atkinson, Ph.D. Second Reader
	Robert Dell, Ph.D. Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Major disruptions such as terror attacks, natural disasters and human failures can have large impacts on critical infrastructure. The rapid reconstitution of those infrastructure systems after a major disruption is crucial to minimize the impact of the disaster. This thesis compares two different modeling techniques to minimize the cost for reconstructing the infrastructure system. The first technique uses a mixed integer linear program to minimize the operation cost of a infrastructure system. The second technique is a graph-based approach in which the vertices of a meta graph represent different operating states for the infrastructure system, and edges between vertices represent possible transitions between states (e.g., the repair of one or more infrastructure components). In this context, optimal restoration of the infrastructure system corresponds to finding the best (e.g., minimum cost) path from an initial damaged state to a fully restored state. We consider two different ways of finding the shortest path in this meta graph, specifically Dijkstra's algorithm and the A-star algorithm. We compare these techniques in terms of quality of solution and required computation time.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Acronyms and Abbreviations

MILP	Mixed Integer Linear Program
DAD	Defender-Attacker-Defender
GAMS	General Algebraic Modeling System

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

Major disruptions such as terror attacks, natural disasters and human failures can have large impacts on critical infrastructure (e.g., telecommunication networks, power networks, transportation networks and distribution networks for water, gas and fuel). It is crucial to restore these critical infrastructure systems to minimize the impact of disasters. The network operator is responsible to find a way of reconstructing the system as fast as possible and while using the least amount of resources. More specifically, the network operator has to balance the time to restore system function, the level of restoration at the end of the time horizon and the amount of resources used during this time horizon. These three things are in tension with another. The network operator can adjust two of them, but the third will be changed in response.

In this thesis, we model an infrastructure system as minimum-cost network flow problem. We compare two different modeling techniques to minimize the operation cost while reconstructing the infrastructure system. We compare these two techniques in terms of computation time to solve this problem and the objective value of the optimal solution. We assume the operation cost of an infrastructure system is the cost for transporting commodities through the system and the penalty cost for not fulfilling demand. The first technique we use is a mixed integer linear program (MILP) with binary decisions representing which repairs to make in each of several time periods. The second technique we use is a graph-based approach to plan the sequence of repairs as a path through the state space of possible repairs. While we reconstruct the infrastructure system we get the optimal flow of the commodities for every time step in the reconstruction horizon. We also get the optimal flow of commodities for the fully functional infrastructure system.

The MILP has decision variables for the flow at every arc at the infrastructure system, the time when to reconstruct an arc and the amount of unfulfilled demand at each node. We have constraints for the balance of flow at each node at each time step, maximal possible flow on each arc at each time step and a constraint for the reconstruction budget per time period.

With our graph-based approach, we build a meta graph where each vertex represents a state

of the infrastructure system and each edge represents a state change. A state change is a reconstruction of an arc in the infrastructure system. After we build the meta graph we need to find the shortest path from the start-vertex to the end-vertex where the infrastructure system is fully functional. We use two different algorithms to find the shortest path. The first algorithm is Dijkstra's algorithm. This algorithm needs the whole meta graph to find the shortest path. The second algorithm we use is A-star. This algorithm builds the meta graph while it is searching for the shortest path. So it does not need the whole meta graph to find the shortest path.

We use a notional infrastructure system to execute reconstruction experiments with different numbers of broken arcs. For each number of broken arcs we choose a random set of broken arcs as the initial state and solve for the optimal reconstruction using all techniques. We measure the computation time to get the optimal solution for all approaches. We see that we need a large amount of computational resources to do the reconstruction experiments, specifically if we have a large number of broken arcs in the infrastructure system.

Given the same number of broken arcs, each technique obtains the optimal solution. This is not necessarily true for bigger, more realistic, and more complex infrastructure systems. More research should be done to compare the MILP with the graph-based approach using both ways to find the shortest path. However, future research will require additional computational resources.

For our notional infrastructure system, it turns out that the MILP is the fastest technique to get an optimal solution for all numbers of broken arcs. The second best was the graph-based approach with the A-star algorithm. We observe that the solution time for all approaches grows exponentially with the number of broken arcs. The results indicate that for larger and more complex infrastructure systems it is possible that the graph-based approach with the system algorithm might outperform the MILP approach. Again, there is a need to do more research for larger and more complex infrastructure systems.

CHAPTER 1:
Introduction

1.1 Backround

Major disruptions to critical civil infrastructure such as terror attacks, natural disasters (Ran, 2011), and human failures can have serious impacts on the humans and the economy of a country. Critical civil infrastructures include telecommunication networks, power networks, transportation networks and distribution networks for water, gas and fuel. The rapid reconstitution of those infrastructure systems after a major disruption is crucial to minimize the impact of the disaster. As described by Lee et al. (2007), when a major disaster strikes, emergency response organizations are activated both within different levels of government in order to minimize human suffering and also within the organizations that operate infrastructure systems to restore system function (Bigley and Roberts, 2001; Scanlon, 1994).

In this thesis, we focus on the decisions of the infrastructure system operator (or simply the *operator*) to restore system function. In general, the operator must balance three things: the time to restore system function (i.e., the *time horizon*), the level of restoration at the end of the time horizon (e.g., 80% of normal function), and the amount of resources used during this time horizon. Essentially, these three things are in tension with one another. The operator can fix any two of these three, but the third will change in response. That is, the operator might be able to get full (100%) restoration quickly, but it might cost a lot of resources. The operator can work with fewer resources, but it will mean either achieving a lower level of restoration over the same time horizon, or it is going to take longer to achieve full restoration.

Throughout this thesis, we follow the basic terminology and notation in Alderson et al. (2014a). We represent an infrastructure system as a network consisting of nodes, links, and transported commodities (e.g., electricity, gas, fuel, water or information). A *node* is a place with either a source or demand of a commodity; a node with zero supply and demand is called a transshipment node. An *link* connects two adjacent nodes. If the link is *functional* it can support the transport of some commodity. A nonfunctional link cannot transport commodities. An *incident*, which could be a terror attack, a natural disaster, or a human failure, results in one or more links becoming nonfunctional. *Reconstruction of the infrastructure system* means restoring the links to make them functional. In general, we assume that all links in the infrastructure system are functional (i.e., the network is *fully functional*) prior to the incident. We also assume that all nodes are functional and stay functional throughout the reconstruction horizon.

We assume that time evolves in discrete time steps. In any given time step, the operator has a finite budget for restoring links. Because of the way in which individual links work in concert to achieve system function, the order of restoration can greatly impact system performance during the restoration process.

1.2 Scope of This Thesis

This thesis considers how best to use limited resources to restore service to a single commodity transportation network following an incident. We use the notional infrastructure system from Alderson et al. (2014b) as a case study. We consider two approaches to to solve this problem. First we use a Mixed Integer Linear Program (MILP). The MILP decides the order in which to restore the system, and how to operate the system during restoration, minimizing the total operating costs over the restoration time horizon.

We consider another conceptual approach based on the changes in functionality of the system as links are restored. More specifically, we define the *operating state* of the infrastructure system in terms of a binary vector of length m, where m is the number of links in the infrastructure system. An element of the binary vector is one if its corresponding link is functional, and it is zero if the corresponding link is nonfunctional. Thus, there are 2^m possible operating states for any infrastructure system.

We then define a meta graph consisting of vertices and edges. Each vertex represents a unique operating state for the infrastructure system and each edge represents a feasible transition between operating states. Each edge is annotated with a value that represents the operating cost associated with the state transition. In this way, the sequence of link repairs corresponds to a sequence of state transitions, which corresponds to a path in the meta graph. On the meta graph, one can define a path-optimization problem where the objective is to find the optimal restoration path from an initial state to a goal state, which represents the fully functional infrastructure system. We refer to this second technique as the graph-based approach to system restoration.

By using both the MILP and the graph-based approach, we consider the following research questions:

- Given a fixed repair budget in each time period, what sequence of repairs results in the shortest repair time?
- Given a fixed repair budget in each time period, what sequence of repairs results in the lowest cumulative operating cost?
- What advantages, if any, does the MILP or graph-based approach have either conceptually or computationally?
- How well does each technique scale to large systems?

The remainder of this thesis is organized as follows. Chapter 2 provides examples of past work related to system reconstitution. Chapter 3 introduces the MILP model for reconstruction, and uses the notional infrastructure system from Alderson et al. (2014b) to perform reconstruction experiments. Chapter 4 details the graph-based approach to solve the reconstruction as a routing problem and presents computational results for the same experiments. Chapter 5 presents conclusions and future work.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Literature Review

The literature on function and restoration of interdependent infrastructure systems is large and growing. This chapter provides an abbreviated chronological review of the work most relevant to this thesis, along with a summary of our contributions.

2.1 Abbreviated Chronological Review

Critical infrastructure systems have become more and more interdependent since the last decade of the 19th century. Therefore, it has become more important to identify, understand and analyze these interdependencies. Key changes in economics and regulations caused the development of highly complex infrastructure with more centralization of control (Rinaldi et al., 2001). Along with it, the restoration process has also become more complex.

Davidson and Cagnan (2003) uses a simulation-based model for post-earthquake restoration processes. They use an electric power system as an example. The goal of their research is to understand the repair and recovery operations of an infrastructure system and to estimate its expected repair time. By using a Monte Carlo Simulation, their model is able to to provide information about the progress at each repair site and utilization of each repair crew.

Rinaldi (2004) states that models and simulations can provide insights into complex critical infrastructure systems. This includes the complex behavior and operational characteristics of interdependent infrastructure systems. Therefore, models must account for interdependencies among infrastructure systems. He observes, that although modeling of interdependencies among infrastructure systems is very challenging, several techniques such as aggregate supply and demand tools, dynamic simulations, agent-based models, population mobility models, Leontief input-output models and physics-based models have been successfully used to model and simulate interdependent infrastructure systems.

Pederson et al. (2006) conduct a survey to identify and catalog research regarding the modeling and analysis of interdependencies of critical infrastructure systems. They find two major approaches which are used to model cross-infrastructure analysis. The "integrated approach" models several infrastructure systems as one big system. The "coupled approach" models each infrastructure system with one individual model and uses the outcome of one model as input for a second model. According to Pederson et al. (2006), integrated models are able to model much higher complexity than coupled models.

Lee et al. (2007) model all interconnections or interdependencies between critical civil infrastructure systems to minimize the impact of disruptions. They use a network flow mathematical representation. They call this an interdependent layer network model. They use the lower Manhattan region of New York for illustration purposes. They show that a model of critical civil infrastructure systems and the interdependencies or interconnections between them can provide powerful means of understanding, monitoring and controlling critical infrastructure systems. They claim that these models are also useful for planing purposes of reconstruction and routing problems.

Gong et al. (2009) combine mathematical programming and constraint programming to schedule the needed tasks for reconstruction of a disrupted system. They use three classic objectives: the cost, the tardiness, and the makespan. They use convex combinations of these objectives to get effective solutions to the multi-objective problem. They obtain a solution by using Benders decomposition for each combination. They split the problem into a master problem and sub-problems. The master problem delivers the solution for an assignment problem, where every task is assigned to a work group. Each sub-problem obtains the solution to a scheduling problem, indicating which task has to be done when. They show that their hybrid decomposition approach is able to solve scheduling problems effectively.

Shoji and Toyota (2009) analyze the restoration process of four interdependent systems of critical infrastructure. These critical systems are electric power supply systems, gas supply systems, water treatments, and communication networks. They present a model based on graph theory to analyze the current operational plans for disaster prevention and the restoration process of critical infrastructures and its interdependencies. They found by sensitivity analysis that there are strong and weak connectivity between various stakeholders involved in the reconstruction process.

Reed et al. (2009) developed a simple model to describe the resilience and interdependency of networked infrastructure while reconstructing these infrastructures. This model does not depict the whole real-time functioning of an infrastructure system. Rather, this model develops strategies to improve infrastructure resilience. They use interdependent power delivery and telecommunication systems as an example to show the capability of their model for the development of design strategies for increased resilience of critical infrastructure.

Matisziw et al. (2010) use a multi-objective linear-integer optimization model to maximize system performance while reconstructing a networked infrastructure over a fixed planing horizon. The two opposing objectives are minimization of the system costs and maximization of the system flow. They use a telecommunication network for illustration purposes. They show the importance of scheduling the restoration of links and nodes simultaneously, instead of incrementally.

Sanchez (2010) uses the example of an Internet Service Provider to show the tensions between changing customer demands and costs to build new infrastructure. He presents three models to support system operation while developing new network infrastructure. The traffic engineering model is constrained to the existing network infrastructure to fulfill the customer demand. The network provisioning model accounts for the ability to build new network infrastructure. The multi-period network provisioning model includes the ability to build new network infrastructure at several time periods. Sanchez (2010) shows that the budget to build new network infrastructure, desired network performance, and planing time horizon have to be in a balance with each other.

Coffrin et al. (2012) present a model for restoring a power network. They split the reconstruction and routing problems into three stages. First, they use a MILP to get the minimum restoration set to restore the network to full capacity. At the second step, they solve the scheduling problem to determine the order in which the network components are restored. They solve the routing problem for every restoration step as third step. Coffrin et al. (2012) focus only on the two first steps. They also use a randomized adaptive decomposition to solve the restoration ordering problem, and they show that the this approach outperforms the MILP approach within given time constraints.

According to Cavdaroglu et al. (2013), restoring a network or public service requires determining the set of components that will be installed and the tasks needed to install these components. Furthermore, one needs to assign the tasks to work groups and generate a schedule for all available work groups. Their model provides an optimal solution for the restoration and planing decisions using a MILP to solve their model. They also account for the interdependencies between several infrastructure systems. Their objective function provides a measure of performance for the network over the horizon of the reconstruction, including how well the network is restored after all restoration steps. They also develop a heuristic solution method, using ideas from network flow and scheduling problems. The heuristic solution method does not need a MILP. Therefore they are able to solve the restoration and scheduling problems with limited computational resources in a reasonable amount of time.

Baxter et al. (2014) use incremental network design problems to illustrate how to find the optimal choice and timing of network expansions. They use a greedy heuristic and show the worst case performance of this heuristic while solving incremental network design problems. Furthermore, they derive an approximation algorithm to solve incremental network design problems.

Álvarez et al. (2014) use a continuous MILP to solve the supply chain network infrastructure problem. They use a road network and try to restore this network after a major disruption. The task for this optimization problem is to assign and schedule repair crews to restore damaged roads (links). They seek to minimize the time for loads and repair crews to get to their destination. Instead of using discrete and a fixed amount of time steps for this optimization, Álvarez et al. (2014) use a variable and continuous time horizon.

Alderson et al. (2014a) use a sequence of models to show how to assess and improve resilience of critical infrastructure. They focus on operational resilience and show how to build and solve these models. First they build an operator model that looks at the network from an operator point of view. The solution of this model represents the usual behavior of the network. Next is the attacker model which takes into account the possible attacks to the network. The solution of the attacker model is an attack which has the greatest negative impact on the performance of the network. The last step is the defender model. The solution to this model is a set of actions (e.g., hardening, redundancy, capacity expansion) which improves the resilience of the network after a worst case attack. The resulting model is a three-stage Defender-Attacker-Defender (DAD) model. Alderson et al. (2014b) use this sequence of models to assess the operational resilience of a notional fuel infrastructure system.

2.2 Contributions in Context

The research in this thesis contributes to this literature in the following ways. We present a MILP formulation that focuses on post-disaster restoration of an infrastructure network with discrete time steps and variable planing horizon. This is similar to the work of Álvarez et al. (2014). Our graph-based approach addresses the same problem by using vertices as operating states and edges to represent transitions between these operating states. The solution to our models shows the order in which we have to reconstruct links of our infrastructure network and also the optimal routing of commodities at every time step to use the least amount of cumulative resources.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Mixed Integer Linear Program Model and Results

In this chapter, we present a technique for solving the infrastructure restoration problem using a MILP model.

3.1 Mixed Integer Linear Program Model

We use the following formulation for a MILP to reconstruct the infrastructure system in an optimal way after a disaster.

Indices and Sets

$t \in T$	time steps, $T = \left\{0, 1, 2, 3, ..., t_f\right\}$, where t_f is the final time step, (alias t')
$n \in N$	nodes (alias i,j)
$[i, j] \in E$	undirected link between nodes i and j, $i < j$; define $m = \lvert E \rvert$
$(i, j) \in A$	directed arc from node i to node j
	$[i, j] \in E \Leftrightarrow (i, < j) \wedge ((i, j) \in A \wedge (j, i) \in A)$

Data [units]

c_{ij}	per unit cost of traversing link $[i, j] \in E$ [dollars/barrels]
u_{ij}	upper bound on total (undirected) flow on link $[i, j] \in E$ [barrels]
d_n	fuel supply at node $n \in N$ [barrels]
	(a negative number for demand, a positive number for supply)
p_n	per unit penalty cost for demand shortfall $n \in N$ [dollars/barrel]
r_{ij}	cost of reconstruction for an link $(i, j) \in A$ [dollars]
b_t	per time step budget for reconstruction for time $t \in T$ [dollars]
λ_t	per time step weight for the operating and penalty costs for $t \in T$

Decision Variables [units]

Y_{ijt}	flow on arc $(i, j) \in A$ at time $t \in T$ [barrels]
S_{nt}	fuel shortfall at node $n \in N$ at time $t \in T$ [barrel]
X_{ijt}	binary indicator 1 if build link $[i, j] \in E$ at time $t \in T$, 0 otherwise

Formulation

$$\min_{X,Y,S} \sum_{t=1}^{t_f} \lambda_t \left[\sum_{[i,j]\in E} c_{ij}(Y_{ijt} + Y_{jit}) + \sum_{n\in N} p_n S_{nt} \right] \tag{3.1}$$

$$s.t. \sum_{(n,j)\in A} Y_{njt} - \sum_{(i,n)\in A} Y_{int} - S_{nt} \leq d_n \qquad \forall n \in N, \forall t \in T \tag{3.2}$$

$$(Y_{jit} + Y_{ijt}) \leq \left(\sum_{t'=0}^{t} X_{ijt'} \right) u_{ij} \qquad \forall [i,j] \in E, \forall t \in T \tag{3.3}$$

$$\sum_{[i,j]\in E} r_{ij} X_{ijt} \leq b_t \qquad \forall t \in T \tag{3.4}$$

$$X_{ijt} \in \{0,1\} \qquad \forall [i,j] \in E \tag{3.5}$$

$$S_{nt} \geq 0 \qquad \forall n \in N, \forall t \in T \tag{3.6}$$

$$Y_{ijt} \geq 0 \qquad \forall (i,j) \in A, \forall t \in T \tag{3.7}$$

Discussion

The objective function (3.1) sums the costs for transportation and costs for penalties for unfulfilled demands. Constraints (3.2) enforce balance of flow at each node in each time period. Constraints (3.3) set the maximal possible flow on each link. The maximal flow a link can support at time t is nonzero only if the link is undamaged or restored by time t. If a link is undamaged at the initial state we set $X_{ij0} = 1$. Constraints (3.4) set a reconstruction budget per time period which can be used for reconstruction of links. Stipulations (3.5) - (3.7) set bounds on the decision variables.

As stated, there is no benefit for saving the reconstruction budget, so the optimization will eventually rebuild all links even if they do not reduce the operating cost. To avoid this behavior, one need only to add the reconstruction cost $\sum_{[i,j]\in E} r_{ij} X_{ijt}$ for every time step to the objective. In this case, there is a one-to-one tradeoff between operating costs and reconstruction costs. Other tradeoffs are possible by adding weights to the costs.

3.2 Notional Fuel Network

We use the notional fuel network shown in Figure 3.1 from Alderson et al. (2014b) and perform some reconstruction experiments, in which we modify the initial state of the notional fuel network and look at system performance while we are restoring the system. We seek a reasonable order for reconstructing the links.

We make several assumptions to make our model sufficiently simple. Our model is based on the assumption that time (t) is discrete. We start the reconstruction process at $t = 1$. At the last time step (t_f) the sum of cost for transportation and cost for unfulfilled demand is minimized. In Figure 3.1 we use numbers next to each node to show the demand $(-d_n)$ at this node. For simplicity, the demand at each node and the penalty for not fulfilling this demand (p_n) are constant over all time steps. We assume that only links not nodes can be destroyed by an incident. We assume the only links we can reconstruct are those shown in Figure 3.1 as dashed lines. All links are undirected, so we are able to put flow on an link in either one of its two directions. We assume that each link reconstruction is complete at the beginning of a time step, and the link can be used for fuel delivery from then on. The flow capacity on link (i, j), denoted u_{ij}, is constant for all time steps. Figure 3.1 shows two numbers on every possible link. The first number is the cost for building (r_{ij}) this link from node i to node j. The second number is the cost for putting one unit of flow (c_{ij}) on this link. The budget per time step to reconstruct links (b_t) is constant for all time steps.

We use the following standard specifications for our reconstruction experiments and refer to them as the base case. The demand at every demand node is one unit of fuel. The supply is always ten units for each of the two supply nodes (black nodes at Figure 3.1). We use $p_n = \$10$ for the penalty cost for unfulfilled demand per unit of fuel per node per time step. Due to the fact that the sum of supplies is bigger than the sum of the demands, we are able to fulfill all demands with the fully operational distribution system. The cost for transporting one unit of fuel along an link is \$1. We set the maximum possible flow for every operational link to 15 units. The cost for building an link is \$5. We fix the reconstruction budget (b_t) to \$9 per time step. So we are able to build not more than one link per time step. Unused reconstruction budget does not accumulate over time steps. We fix the weight factor (λ) for the sum of transportation costs and penalty costs to one for all time steps. After full reconstruction, the infrastructure system is again fully operational,

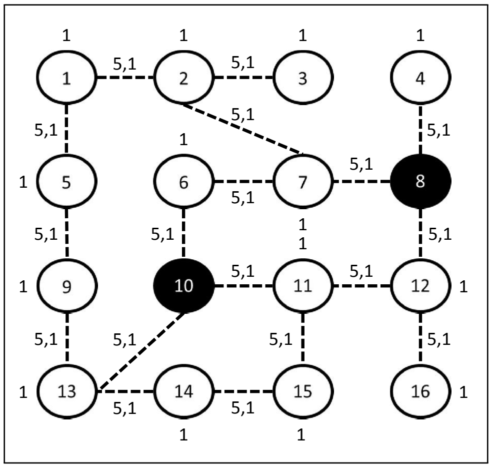

Figure 3.1: Notional Infrastructure Network with possible links (dashed lines), cost for building link (first number at link), transport cost per unit flow over one link (second number at link) and demand at nodes (numbers next to each node).

works at optimal transportation cost and all demands are satisfied.

We use the operating costs for fuel delivery (including penalties for non-delivery) as a measure of performance for the infrastructure system. The operating costs are the sum of costs for transportation, costs for building new links and penalty costs for not fulfilling demand. The system performance is measured at every time step during the whole reconstruction process and we optimize the sum of the performance over all time steps.

3.3 Implementation

We implement the MILP in the General Algebraic Modeling System (GAMS) (GAMS Development Corp., 2014) and solve it using the CPLEX solver. For the experiments here, there are $m = 18$ links and $t_f = 18$ time steps. This results in 324 binary variables and 936 continuous variables. We solve all problem instances in this chapter on a laptop computer with an Intel i3 processor and 8 gigabytes of memory. Depending on the number of broken links that need to be rebuilt, the runtimes vary from a few seconds to approximately 4.5 hours.

3.4 Results from MILP Model

In this section we show the results from two different reconstruction experiments. First, we reconstruct a fully destroyed infrastructure system and second we reconstruct a partly destroyed infrastructure system.

3.4.1 Reconstruction of a Totally Destroyed Infrastructure System

Our first experiment is to reconstruct the fully destroyed infrastructure system (i.e., a network with no operational links). For this experiment we use all assumptions and system parameters from the base case. We show the optimal order in which to build the edges of the system in Figure 3.2. According to the optimal solution of the MILP, we build all necessary links starting from the two source nodes to the directly adjacent demand nodes. This takes from time step $t = 1$ to time step $t = 6$. The order we use to build the first six links is arbitrary. A different order for the first six reconstructions does not change the cumulative system performance. Then, we build the edges necessary to fulfill the demand at nodes that are two transportation steps away from the source nodes. This lasts from time step $t = 7$ to time step $t = 11$. Finally we build links necessary to fulfill the demand at the remaining nodes which are three steps away from the source nodes. At an optimal solution, it is not necessary to transport units of fuel more than three transportation steps to fulfill all the demands at our notional infrastructure system. The flow at the fully reconstructed system is the same as the flow found by Alderson et al. (2014b). Figure 3.3 shows the optimal flow for the fully restored infrastructure system.

In general, if unsatisfied demand penalties are the same for all nodes, the order in which we restore node connectivity is the same as the order of transportation costs to the nodes.

Figure 3.4 shows the cost to operate this system in each time step $t = 0$ to $t = 18$. The operation cost decreases at every time step, which means the system performance increases at every time step. The operating cost decreases in an approximately linear way. The absolute value of the slope gets smaller, while the transport distance between supply node and reconstructed link increases.

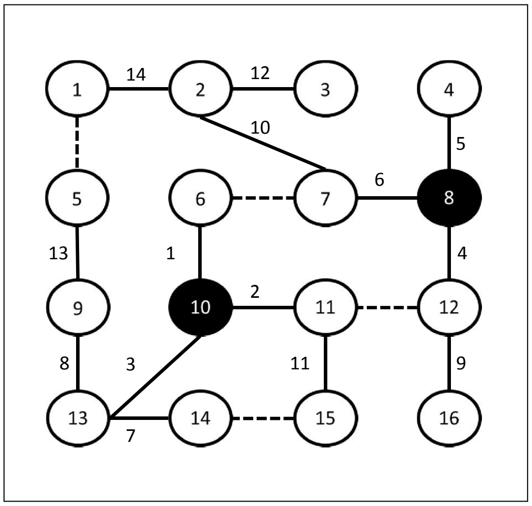

Figure 3.2: Link labels indicate the optimal order of building the notional infrastructure system.

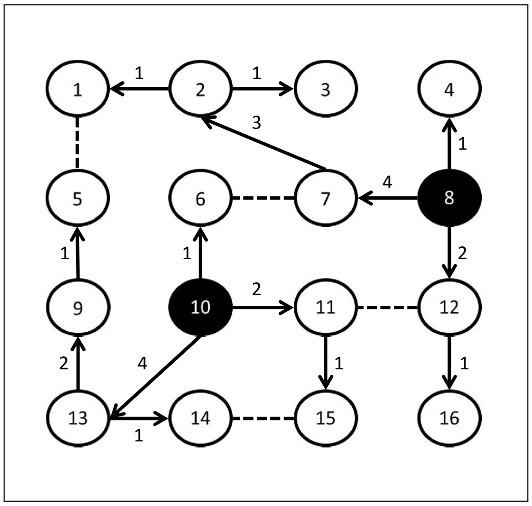

Figure 3.3: Optimal distribution for fully operational notional infrastructure system with indicated direction and amount of flow for each link.

3.4.2 Reconstruction of a Partly Destroyed Infrastructure System

In our next experiment, we consider the reconstruction of a partially destroyed infrastructure system, specified in terms of an initial operating state. We consider the solution from our MILP formulation for several scenarios.

First we assume that our system is in the operating state as shown in Figure 3.5. We change the reconstruction cost at the link between node eight and node seven to \$5. We also change the cost for transportation one unit of fuel along this link to \$5. All links at the left side of our system are operational. The cost for building these links is \$0. The cost for transporting one unit of fuel along one of these links is \$1. Thus, the initial state of the network has some undamaged links. However, link $[7,8]$ is more expensive to rebuild and more expensive to use.

Figure 3.6 shows the order of reconstruction of links. As we can see the solution of the MILP indicates that it is best to reconstruct link $[10,13]$ at first to use the operational links on the left side of the system. This single reconstruction decreases the operation cost dramatically as shown in Figure 3.8. The order or reconstructing links in time steps two to eight follows the rules discovered in section 3.4.1. At time step $t=9$ we build a loop and are able to reroute the fuel units to get a better system performance. We cannot decrease

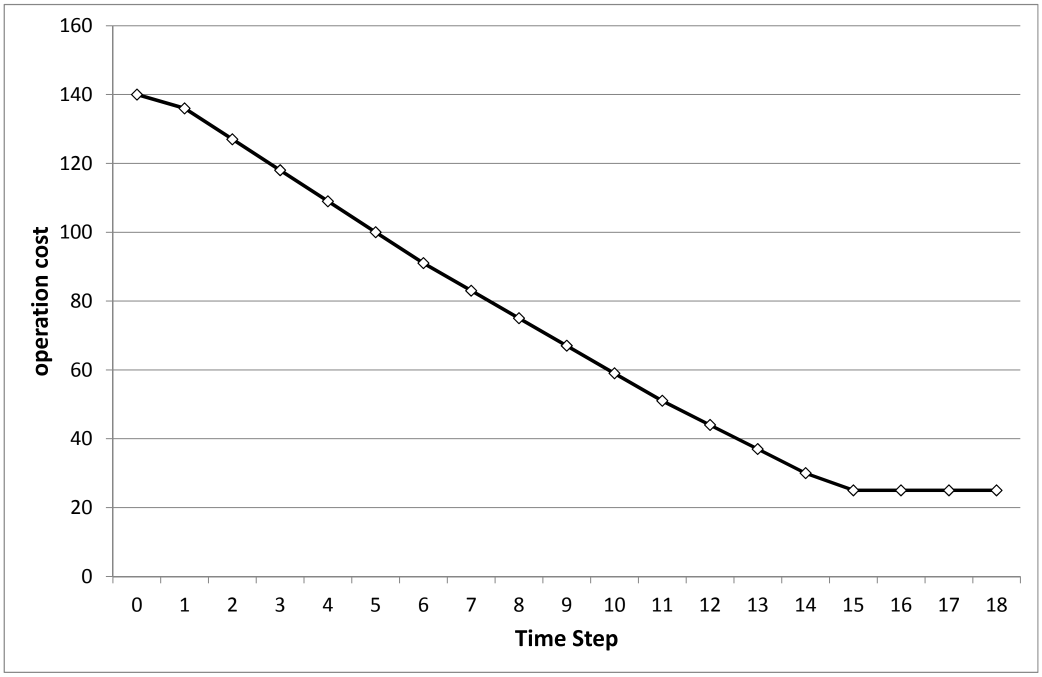

Figure 3.4: Operation cost per time step over time steps.

the operation cost at time step $t = 10$, because we use all available units of supply from node 10. However, we build link $[2,3]$ to connect node 3 with our system. At the next time step, we build link $[11,12]$ to be able to get units of supply from node 8. Now, we can fulfill all demand at our system. The decrease of the operating cost from time step $t = 11$ to time step $t = 12$ can be explained with the fact that we are not building any links later than time step $t = 11$.

Figure 3.7 shows the optimal flow of commodities at the fully operational infrastructure system from Figure 3.5. At an optimal solution it turns out that link $[1,5]$ is not used for transportation of commodities. This is due to the distance of transportation from supply node 10 to demand nodes 1 and 5. Link $[7,8]$ is not built and used because of the high building and transportation costs for this link.

At another experiment we a scenario where the infrastructure system has again two undamaged links. We observe that the solution for the MILP indicates that we must utilize unbroken links as soon as possible, as shown in Figures 3.9-3.11. The first link we rebuild is the link $[10,11]$ to utilize the undamaged link $[11,12]$. Then, we build link $[7,8]$ to uti-

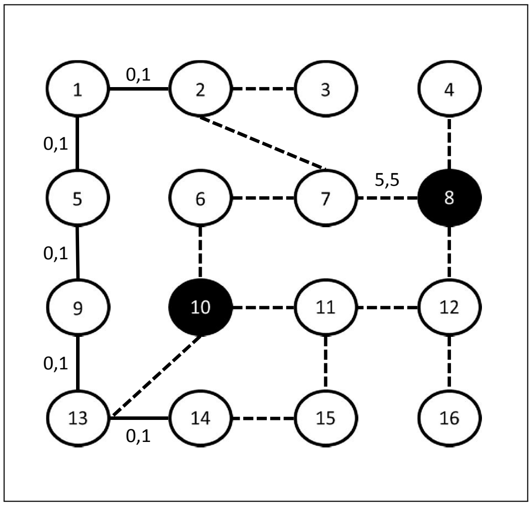

Figure 3.5: Network state after incident and at the beginning of reconstruction. Solid lines represent functional links, while dashed lines represent damaged links that require reconstruction. All system parameters that are not shown are the same as in the base case.

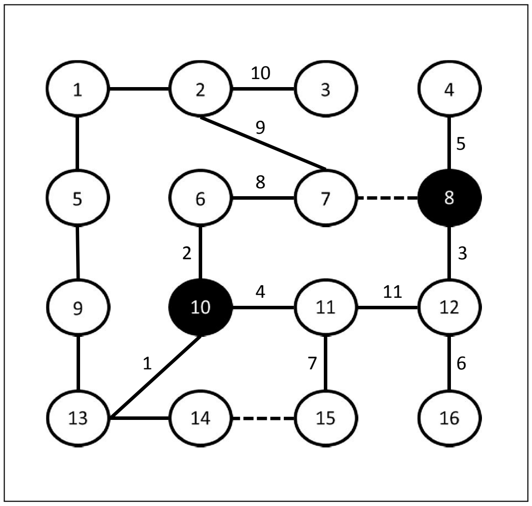

Figure 3.6: Order of reconstructing links for the partly destroyed infrastructure system in Figure 3.5.

lize the undamaged link $[6,7]$. The first fife time steps of the reconstruction are used to utilized undamaged links. This is not surprising, because if we reconstruct an link to utilize another undamaged link, we can fulfill the demand of two nodes. We have to pay the cost for reconstructing one link, but at the same time can fulfill the demand at two nodes. This reduces the penalty cost for unfulfilled demands by $2p_n$.

We repeat this analysis for several different scenarios, each specified by a different initial state, and possibly with different operating costs, reconstruction costs, and/or reconstruction budgets. Details of these scenarios and their results appear in the Appendix.

In general we observe two different phases for reconstructing the notional infrastructure system. There is a first phase where we try to fulfill all demands as soon as possible and a second phase where we optimize the flow of commodities through the infrastructure system. At the first phase we reconstruct the broken links in an order to utilize operational links for transportation of commodities. This reduces the penalties for unfulfilled demand, and we observe a rapid increase in the performance of the system. At the second phase, we optimize the cost for transportation for the commodities through the infrastructure system. Due to the relatively low transportation cost at our notional infrastructure system the

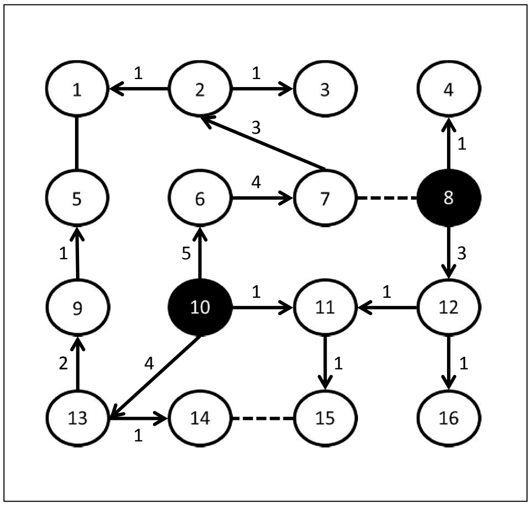

Figure 3.7: Flow of commodities thru notional infrastructure system shown in Figure 3.5 after full reconstruction.

increase in the performance in the last steps of reconstruction is relatively small in comparison to the performance increase at the beginning of the reconstruction. We clearly see that the first few steps are used to fulfill as much demand as possible and the last few steps are used to reduce the transportation coast at the infrastructure system. During the intermediate reconstruction steps we see a mixture of both phases. The transition from the first to the second phase is continuous.

3.4.3 Discussion

From our analysis, we observe that the MILP is an effective technique to identify the optimal restoration of a damaged infrastructure system (represented as a network flow model). However, a disadvantage of the MILP is that it assumes that there will be no surprises (i.e., additional breaks) during the reconstruction and that all model parameters (e.g., costs and durations) are fixed and known. Should something unexpected occur, we need to start the whole computation process allover again. The next chapter considers a graph-based approach which is capable of adapting to surprises as they occur.

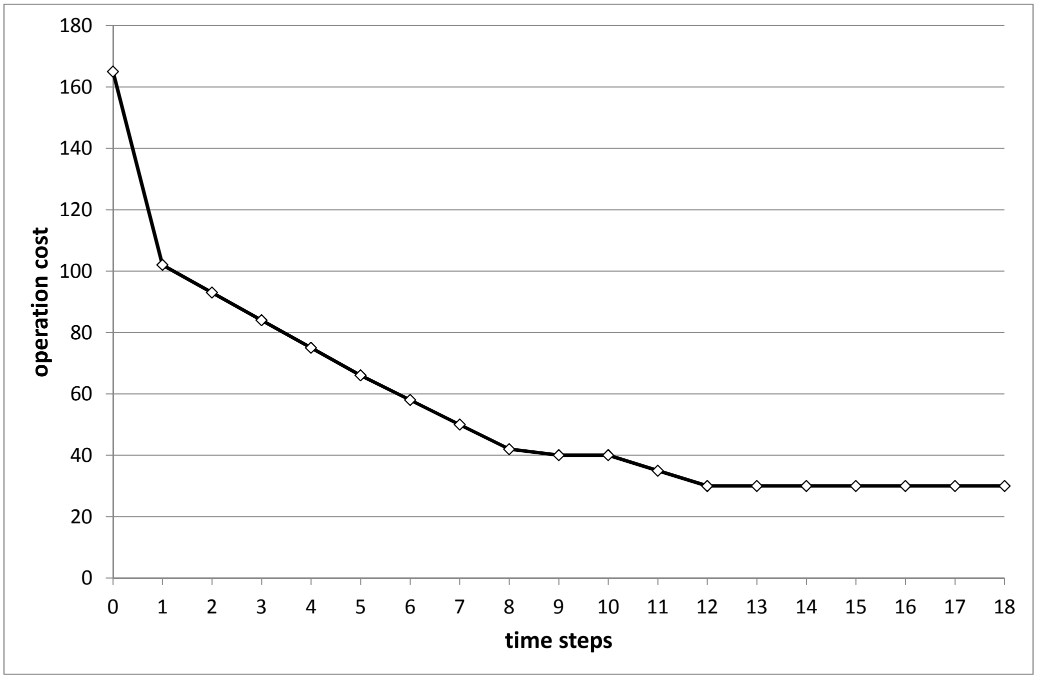

Figure 3.8: Operation cost per time step for case 8 (partly destroyed system).

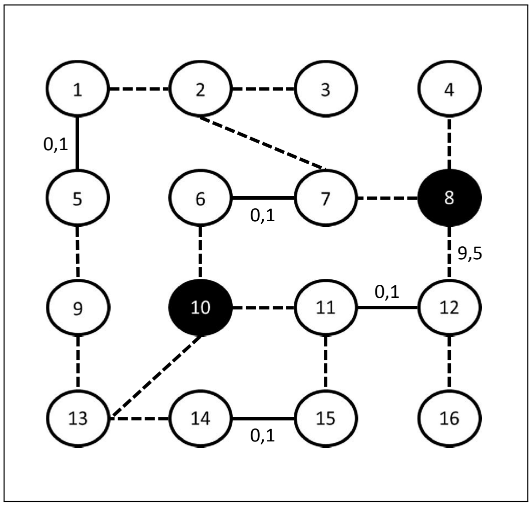

Figure 3.9: Network state after incident and at the beginning of reconstruction. Solid lines represent functional links, while dashed lines represent damaged links that require reconstruction. All system parameters that are not shown are the same as in the base case.

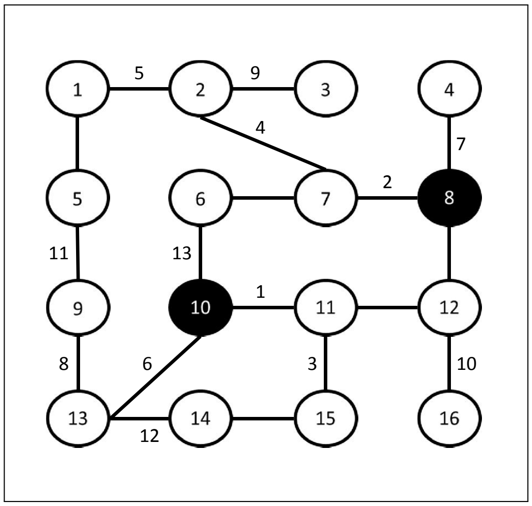

Figure 3.10: Link labels indicate the optimal order of building the notional infrastructure system in Figure 3.9.

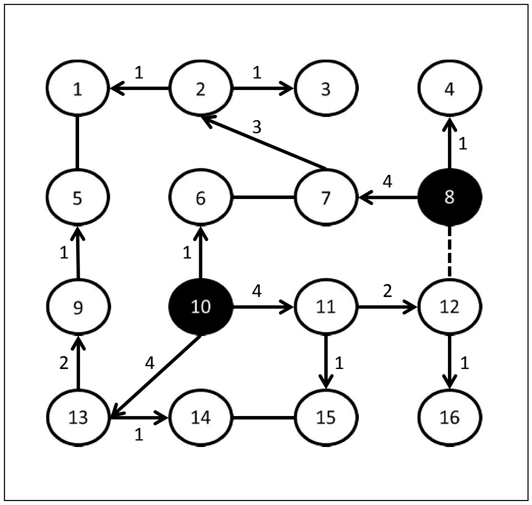

Figure 3.11: Optimal distribution for fully operational notional infrastructure system in Figure 3.9 with indicated direction and amount of flow for each link.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Graph-Based Approach and Results

4.1 Meta Graph

In this chapter, we present another analysis technique in which we build a meta graph that represents the different reconstruction states of the infrastructure system. Each vertex in the meta graph represents one reconstruction state of the infrastructure system. The edges of the meta graph represent possible state changes from one state to another state of the infrastructure system.

Figure 4.1 shows a sketch of the meta graph. The Start vertex S represents some initial state of the infrastructure system (e.g., at time 0). The End vertex E represents the fully operational infrastructure system, where every link of the infrastructure system is built and the system operation (i.e., routing of the commodities) is optimal. As shown in Figure 4.1 we index all vertices with two numbers. The first number is the time step in which this vertex is reached and the second number is the relative order of the vertices within a time step. We associate several pieces of information about the infrastructure system with each vertex of the meta graph. We associate the state, functional or not functional, of every link of the infrastructure system along with the operation cost of the infrastructure system at state S. We use the operator model as it is described by Alderson et al. (2014a) to find the optimal routing of the commodities through the infrastructure system. This operator model gives us the operation cost for the infrastructure system at state S.

We define the length of an edge from vertex S to vertex S' to be the operation cost $P(S)$ of the infrastructure system at vertex S. A change to the state of the infrastructure system corresponds to a move in the meta graph.

Given a meta graph built in this manner, reconstruction of the infrastructure system means finding a (shortest) path in the meta graph from Start vertex S to End vertex E.

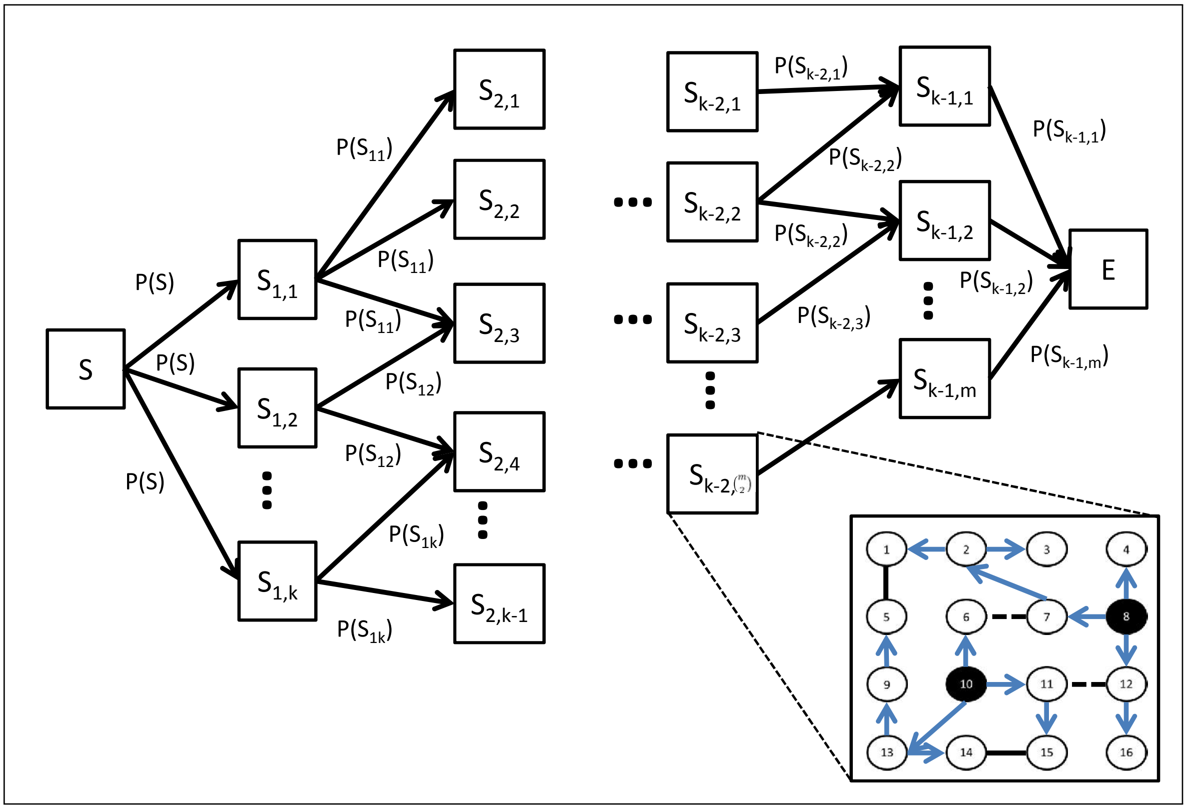

Figure 4.1: Meta Graph with nodes as representation of states of the infrastructure system and edges as representation of state changes between states of the infrastructure system.

4.2 Finding the Shortest Path

In this section we discus Dijkstra's algorithms and the A-star algorithm to get the shortest path from the Start node S to the End node E.

4.2.1 Dijkstra's Algorithm

Dijkstra's algorithm described by Dijkstra (1959) is an algorithm to find the shortest path between two nodes S and E. The algorithm uses the fact that if a node S' is in the shortest path from S to E, then the shortest path from S to S' is part of the shortest path from S to E. The algorithm divides the nodes into two sets (unvisited and visited) and associates a distance label with each node. Distance labels of nodes can be updated until they are marked as visited. In each iteration, the algorithm marks a single node as visited.

We seek the shortest path from the Start vertex S to the End vertex E in the meta graph.

To apply Dijkstra's algorithm, first we need to build the entire meta graph and compute the performance for every state of the underlying infrastructure system. Because we have 2^k possible states in the infrastructure system, we have 2^k vertices in the meta graph, where k is the number of broken links at the infrastructure system. After the whole meta graph is built we use Dijkstra's algorithm to get the shortest path from S to E.

Due to the huge number of nodes, the time we need to build the meta graph can be very long. We need to compute the performance of each vertex in the meta graph, more specifically we need to solve the operator model for each state in the underlying infrastructure system. Even if each computation to solve the operator problem takes a small amount of time, we have to do this 2^k times for the meta graph. At bigger and more complex infrastructure systems the time to build the meta graph can exceed the time we need to find the shortest path in the meta graph by far.

For our reconstruction experiments, we use the following steps to achieve the optimal solution. First we choose k nonoperational links at random from our notional infrastructure system. Then we built the meta graph and compute the performance at each vertex. At least we seek for the shortest path from the Start vertex to the End vertex.

The next section describes how to use the A-star algorithm to find the shortest path in the meta graph. This algorithm needs to solve only a subset of 2^k operator problems to find the shortest path in the meta graph.

4.2.2 A-star

The A-star algorithm was first described by Hart et al. (1968). As described by Dechter and Pearl (1985), A-star uses a general best-first strategy to find the shortest path from node S to node E. Each node S' gets placed in a priority queue according to the sum of the already computed path length from node S to node S' and an estimation of the remaining path length to node E. The node S' with the smallest value is placed at the top of the queue. The neighbors of this node are the next nodes where the actual path and the estimation of the remaining path are computed. We need to repeat this until we get the path length to node E and all other sums of computed path and estimated remaining path length are higher. The A-star algorithm returns the length of the shortest path from node S to node E. To get the path which has this length we need to add another step where each node knows her predecessor. Now we can go backwards from node E to node S to get the shortest path.

We use the A-star algorithm to get the shortest path from the Start vertex S to the End vertex E in the meta graph. To apply the A-star algorithm we need to compute a lower bound of the path length from the Start vertex S to the End vertex E. Assume, we have computed the path length from the Start vertex S to the vertex S'. The lower bound of the path length from the Start vertex S to the End vertex E is the summation of the path length we have computed so far and an estimation of the remaining path length to vertex E. We use a relaxation of the MILP model to get the estimation of the remaining path length. For this relaxation we relax the binary constraint for the decision variable X to be non-negative real. So we can build parts of links at the infrastructure system. We also relax the non-negative integer constraint for the Y to non-negative real. So we can transport parts of commodities through the infrastructure system. Both relaxations speed up the time for solving this optimization problem and we get a lower bound for the remaining path length relatively quick.

The A-star algorithm explores the meta graph while building it. The algorithm uses regional information (e.g., the estimation of the remaining path to the end vertex) to decide which path has the highest potential to be the shortest path and it explores only a subset of the meta graph. Because the A-star algorithm explores only a subset of the meta graph, this algorithm is needs to solve only a subset of the operator problems. But in addition it needs computation time to get a estimation of the remaining path.

The A-star algorithm solves the operator problem and computes an estimation of the remaining path for a subset of the vertices in the meta graph. For our reconstruction experiments with the A-star algorithm we use the same randomly chosen broken links as in the experiments with Dijkstra's algorithm. The A-star algorithm explores only parts of the meta graph and returns the shortest path from the Start vertex S to the End vertex E.

4.2.3 Comparing Dijkstra and A-star

There is now a large literature on the comparison of various shortest path algorithms and their performance (e.g., Gallo and Pallottino, 1988). In general, Dijkstra's algorithm can be viewed as a special case of the A-star algorithm where the estimation of the remaining path length is zero. Although using a nonzero estimate of the remaining path length makes the A-star algorithm somewhat heuristic, this additional information can give A-star better pathfinding performance for many problems. As long as this estimate is a lower bound on the remaining path length, the A-star algorithm is guaranteed to achieve the correct answer.

In our computational experiments, the comparison is not strictly between Dijkstra's algorithm and the A-star algorithm, but also includes the computational time required to build the meta graph itself. The hope is that by not having to build the entire meta graph, the A-star algorithm achieves shorter runtimes.

4.3 Computational Experiments

In this section we present the implementation and results of our computational experiments.

4.3.1 Implementation

We implement the experiments in this chapter using the Python programming language (Python Software Foundation, 2014). To construct the meta graph we use the NetworkX library (NetworkX Developer Team, 2014). We use the NetworkX built-in implementation of Dijkstra's algorithm, and implement the A-star algorithm ourselves. We solve the relaxation of the MILP using the Pyomo optimization library (Hart et al., 2012) with the CPLEX solver.

For comparison purposes, we also implement the MILP in Pyomo with the CPLEX solver. All computational experiments were conducted on a desktop computer with an Intel Xeon 2.13GHz processor and 48 gigabytes of memory.

For the infrastructure system under consideration here, there are $m = 18$ links, resulting in a meta graph with $2^m = 262,144$ vertices.

4.3.2 Results

Figure 4.2 shows the time needed for solving the reconstruction problem for the different approaches over the number of broken links at the notional infrastructure system. We vary the number of broken links from 1 to 18, which is the maximal number of possible links at our infrastructure system. Due to the fact that we need to build the whole meta graph to use Dijkstra's algorithm, this algorithm is the least efficient way we have seen in our experiments. The A-star algorithm does not require the entire meta graph and should be faster than Dijkstra's algorithm. The MILP is the approach which needs the least amount

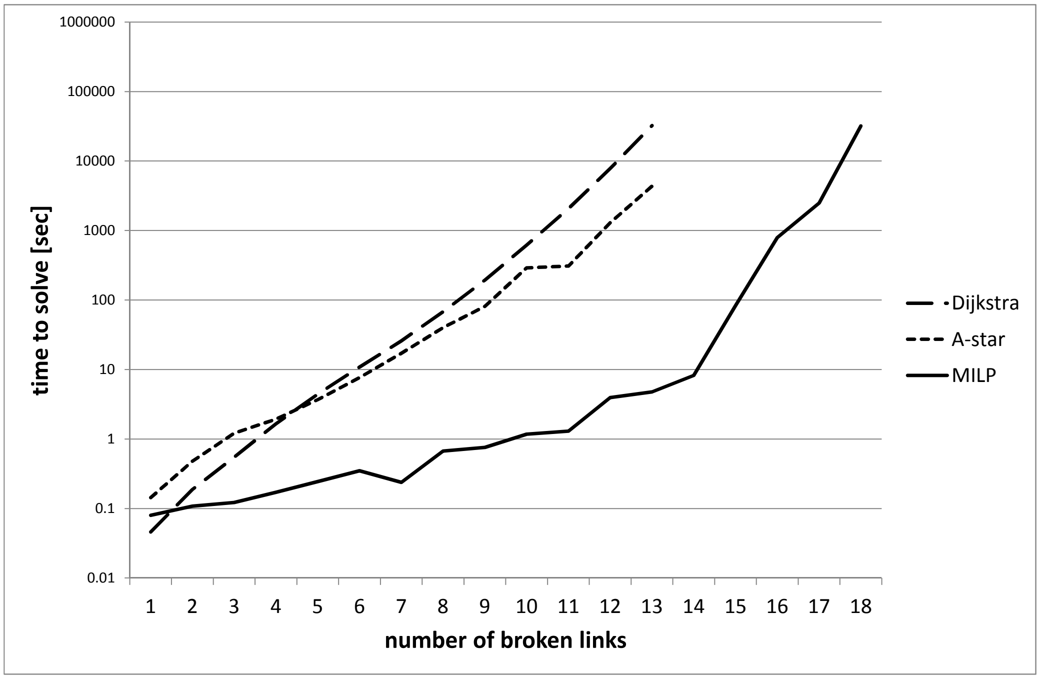

Figure 4.2: Time to get the optimal solution for different approaches over number of broken links.

of time to solve the reconstruction problems. This result is not surprising, because the algorithms for solving MILP are highly developed and well implemented into available solvers over the last 30 years.

We observe that the MILP model is faster than the graph-based approach with Dijkstra's algorithm for finding the shortest path. Only in the very simple case of one broken link the graph-based approach with Dijkstra's algorithm is faster than the MILP. The routines to solve MILP are highly developed and need a short time to solve. If we use the MILP approach to get the optimal solution for our reconstruction problem, we have to solve only one MILP. But, if we use the graph-based approach we need to solve 2^k operator problems, which are small MILPs.

For small k the A-star algorithm needs about the same solution time as Dijkstra's algorithm as shown in Figure 4.2. We can explain this behavior in the following way. The A-star algorithm explores less vertices in the meta graph, but needs more computation time per vertex than Dijkstra's algorithm. The effect of compute the performance and the estimated remaining path length dominates the effect of exploring less vertices at the meta graph for a small numbers of broken links.

If the number of broken links at the infrastructure system is relatively big the effect of exploring less vertices in the meta graph dominates the effect of compute the performance and the estimated remaining path length. The graph-based approach with the A-star algorithm is significant faster than the graph-based approach with Dijkstra's algorithm only for big numbers of broken links.

We see in Figure 4.2 that the solution time for Dijkstra's algorithm and the A-star algorithm are growing with the same rate for small numbers of k. But, we see that the difference between the two algorithms gets significant for bigger k; here the A-star algorithm solves the problem faster than Dijkstra's algorithm. This suggests for even larger and more complex infrastructure system that the A-star algorithm will outperform Dijkstra's algorithm for finding the optimal solution.

The MILP needs significantly less time to solve the reconstruction problem than the graph-based approach with the A-star algorithm for all values of k. But, Figure 4.2 shows that the solution time for the MILP increases rapidly for bigger values k. This observation suggests for larger and more complex infrastructure systems that there is a point where the solution time of the MILP could exceed the solution time of the graph-based approach with the A-star algorithm.

In our experiments we assumed that we cannot destroy any links while we reconstruct the infrastructure system. If we relax this assumption and an link gets destroyed in the reconstruction horizon the MILP needs to be started all over again. If we use the graph-based approach and an link gets destroyed we can use the same meta graph. All computations we have done so fare are still valid and can be used to find the shortest path from our new Start vertex S' with the broken link to the End vertex E.

The ability to speed up the MILP is very limited due to the fact that so much research has been done at this approach. As described in section 4.2.2 we use a relaxation of the MILP model to get the estimation of the remaining path length to the end vertex E. If we are able to get a better estimate for the remaining path in a reasonable amount of time the A-star algorithm can be sped up. Another way to speed up the solution time is to use an algorithm which explores even fewer vertices of the meta graph but still gives us the shortest path from the Start vertex A to the End vertex E.

CHAPTER 5:
Discussion and Future Work

This thesis addresses the problem of reconstructing a damaged infrastructure system by using two different approaches. We consider a MILP and a graph-based approach to get the optimal solution for this problem. Both approaches give us a solution for the minimal cumulative operation cost of the infrastructure system, the optimal order of reconstructing links at the infrastructure system, and the optimal flow of commodities at every time step within the reconstruction horizon.

For the graph-based approach we research two different algorithms to get the shortest path from the Start state to the End state. We use Dijkstra's algorithm where we need to build the whole meta graph to get the optimal solution. The second algorithm we use is A-star, where we need to explore only a subset of the meta graph to get the optimal solution.

We found that the MILP was by far the fastest approach. However, a big disadvantage of this approach is, if one or more links break while we reconstruct our network, we need to restart the whole computation all over again. This approach is not able to handle dynamic changes.

The graph-based approach with Dijkstra's algorithm is capable of handling dynamic changes. Specifically, if one or more links in the infrastructure system break while we reconstruct the network, we just need to restart at a new vertex which represents this new state of the infrastructure system and go on with the computation. All computed values at the meta graph are still valid. The down side of using Dijkstra's algorithm is the large amount of needed computation time while we build the entire meta graph.

The graph-based approach with the A-star algorithm is also capable of handling dynamic changes. This algorithm needs more computation time per vertex of the meta graph but it explores fewer nodes than Dijkstra's algorithm. The positive effect of exploring fewer nodes is significant at larger numbers of broken links. So for large numbers of broken links the A-star algorithm is faster than Dijkstra's algorithm. But, even the graph-based approach with the A-star algorithm is slower than the MILP.

We use a very simple notional infrastructure system for our reconstruction experiments, so future work could explore bigger and more complex infrastructure systems and compare both approaches. We also need to simulate breaking links while reconstructing to see how the different approaches behave in a more realistic dynamic environment. To get a more realistic behavior of the infrastructure system we need to add dependencies in reconstruction (e.g., a particular link has to be reconstructed before starting reconstruction at another particular link) and compare both approaches.

Another interesting point to explore is how both approaches behave if we add more complexity to the reconstruction process. Specifically, what happens if we add dynamic reconstruction budgets? For a more complex and realistic view we can add repair crews with a crew schedule. Adding the crew schedule problem to the computational task would increase the complexity considerably. To drive this idea of a more realistic reconstruction problem further we might add travel distances for the crews as well as travel times and setup costs. The complexity of the reconstruction model is only limited by the available computational resources or the amount of time at hand to solve this problem.

APPENDIX: Reconstruction Experiments

A.1 More Expensive Rebuild and Transportation Costs

In the experiment shown at Figures A.1-A.3, we increase the cost for building links $[2,7]$ and $[7,8]$ to \$9. We also increase the cost for transportation to \$5 per unit of commodities at links $[2,7]$ and $[7,8]$. Link $[5,9]$ is operational at the beginning of our reconstruction experiment. We force the infrastructure system to use all available commodities from the source node 10.

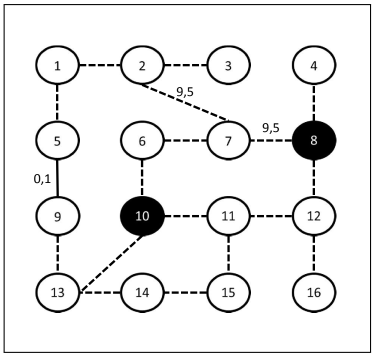

Figure A.1: Network state after incident and at the beginning of reconstruction. Solid lines represent functional links, while dashed lines represent damaged links that require reconstruction. All system parameters that are not shown are the same as in the base case.

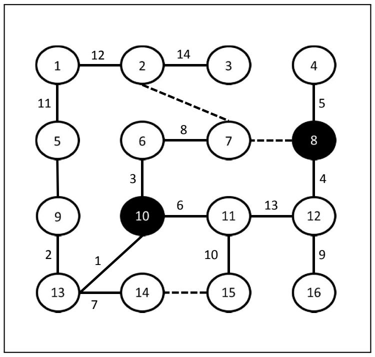

Figure A.2: Link labels indicate the optimal order of building the notional infrastructure system in Figure A.1.

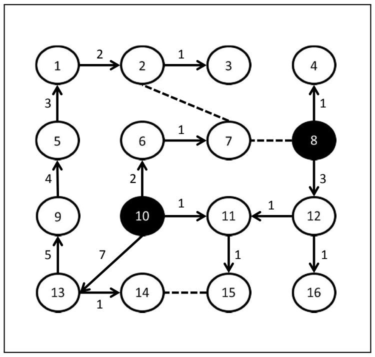

Figure A.3: Optimal distribution for fully operational notional infrastructure system in Figure A.1 with indicated direction and amount of flow for each link.

A.2 Force Not to Use a Particular Link

As you can see in Figures A.4-A.6, we force the infrastructure system to not use the link $[10, 13]$. We see an interesting result where the demand of node 9 is fulfilled from the source node 8.

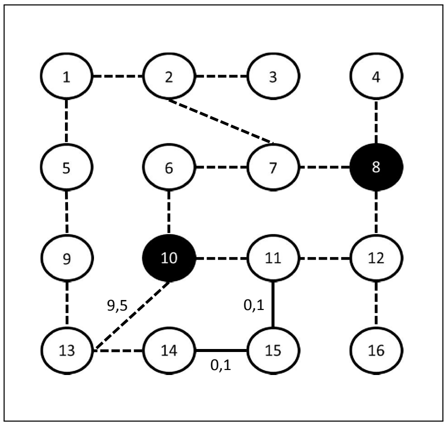

Figure A.4: Network state after incident and at the beginning of reconstruction. Solid lines represent functional links, while dashed lines represent damaged links that require reconstruction. All system parameters that are not shown are the same as in the base case.

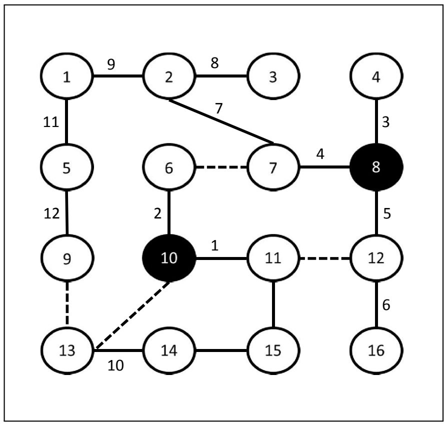

Figure A.5: Link labels indicate the optimal order of building the notional infrastructure system in Figure A.4.

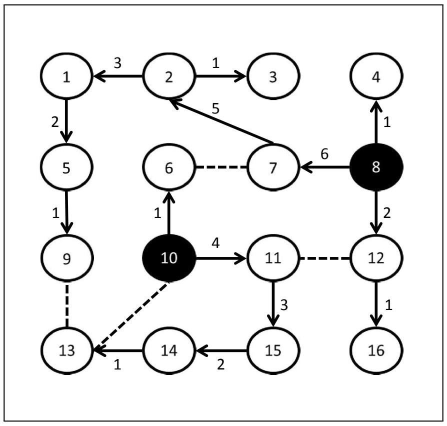

Figure A.6: Optimal distribution for fully operational notional infrastructure system in Figure A.4 with indicated direction and amount of flow for each link.

A.3 Utilize Unbroken Links

In the experiment shown in Figures A.7-A.9, we research how to utilize unbroken links. The solution suggests instead of building link $[1,2]$, we should use the already functional links $[5,9]$ and $[9,13]$.

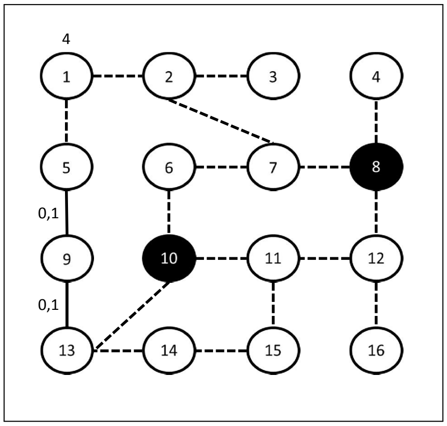

Figure A.7: Network state after incident and at the beginning of reconstruction. Solid lines represent functional links, while dashed lines represent damaged links that require reconstruction. All system parameters that are not shown are the same as in the base case.

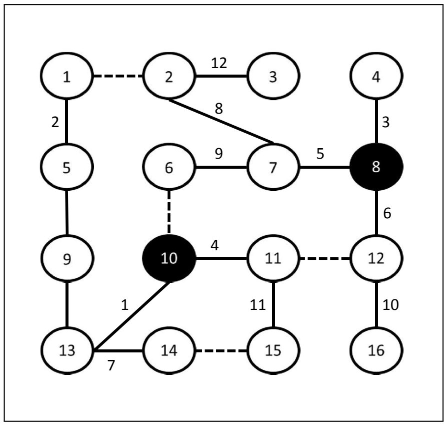

Figure A.8: Link labels indicate the optimal order of building the notional infrastructure system in Figure A.7.

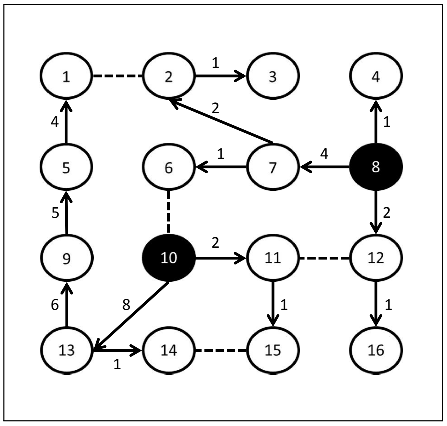

Figure A.9: Optimal distribution for fully operational notional infrastructure system in Figure A.7 with indicated direction and amount of flow for each link.

A.4 High Demand with Unbroken Links

At experiment shown in Figures A.10-A.12 it is possible to fulfill the demand at node 4 from source node 8 and/or source node 10. The solution from the MILP suggests to use node 8 for fulfilling the demand at node 4, because the length of the transport way is shorter.

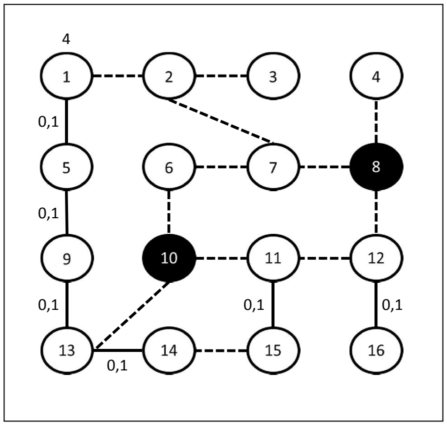

Figure A.10: Network state after incident and at the beginning of reconstruction. Solid lines represent functional links, while dashed lines represent damaged links that require reconstruction. All system parameters that are not shown are the same as in the base case.

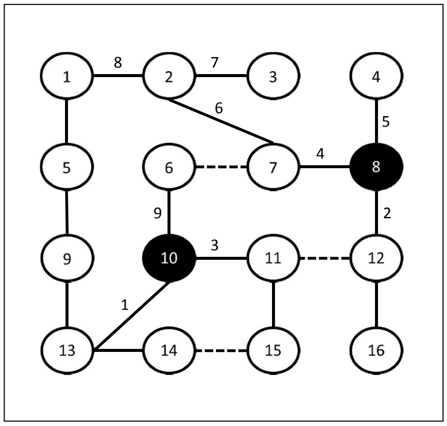

Figure A.11: Link labels indicate the optimal order of building the notional infrastructure system in Figure A.10.

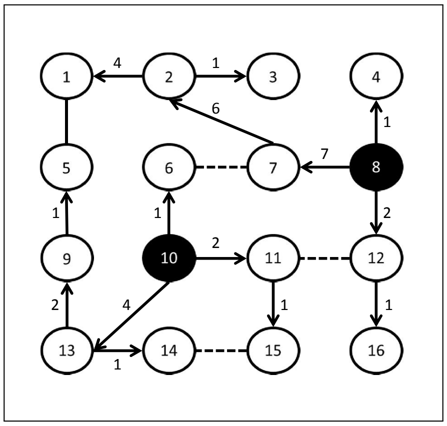

Figure A.12: Optimal distribution for fully operational notional infrastructure system in Figure A.10 with indicated direction and amount of flow for each link.

A.5 Use Supply from a Particular Node

We force the infrastructure system to not use link $[10, 11]$ by increasing building and transport costs at this link. The system reacts as expected, the demand for nodes 11 and 15 are fulfilled from source node 8.

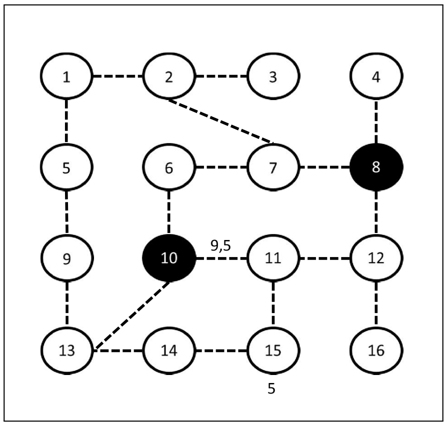

Figure A.13: Network state after incident and at the beginning of reconstruction. Solid lines represent functional links, while dashed lines represent damaged links that require reconstruction. All system parameters that are not shown are the same as in the base case.

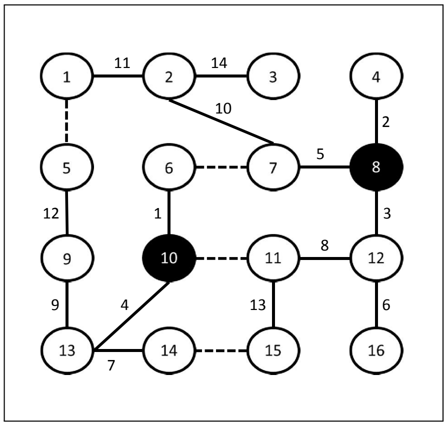

Figure A.14: Link labels indicate the optimal order of building the notional infrastructure system in Figure A.13.

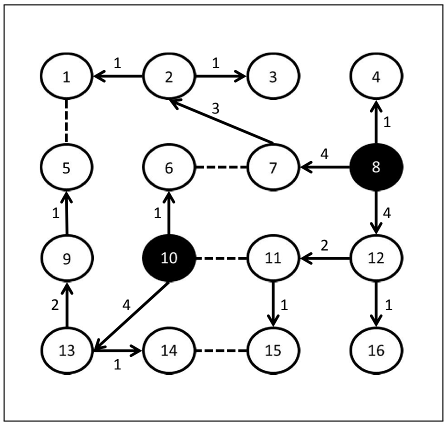

Figure A.15: Optimal distribution for fully operational notional infrastructure system in Figure A.13 with indicated direction and amount of flow for each link.

A.6 High Demand with High Building and Operation Costs

At Figures A.16-A.18 we see a high demand close to source node 10 as well as high building cost at link $[10,13]$. The solution from the MILP suggest to build this expensive link right at the beginning of the reconstruction horizon. If we reconstruct the links as suggested, we can fulfill the high demand at node 5 at the second step and decrease the penalty cost by a large amount.

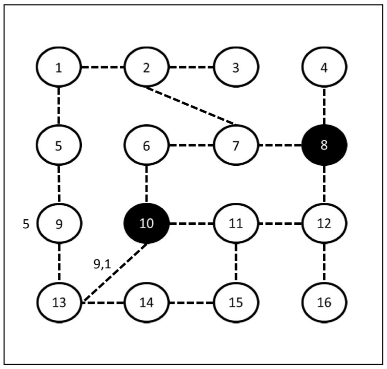

Figure A.16: Network state after incident and at the beginning of reconstruction. Solid lines represent functional links, while dashed lines represent damaged links that require reconstruction. All system parameters that are not shown are the same as in the base case.

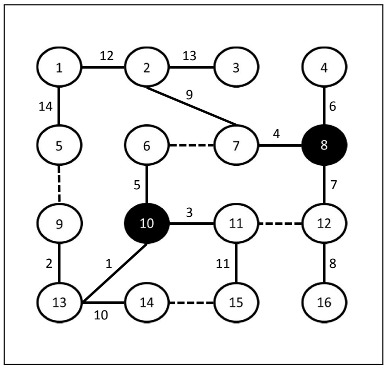

Figure A.17: Link labels indicate the optimal order of building the notional infrastructure system in Figure A.16.

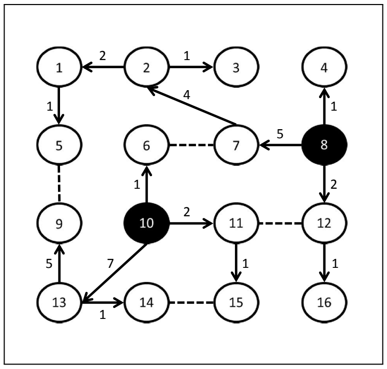

Figure A.18: Optimal distribution for fully operational notional infrastructure system in Figure A.16 with indicated direction and amount of flow for each link.

A.7 Build a Link at a Certain Time Step

In the experiment shown in Figures A.19-A.21 we increase the cost for building link $[4,8]$. We see that all other links which start at a source node are build previous to link $[4,8]$.

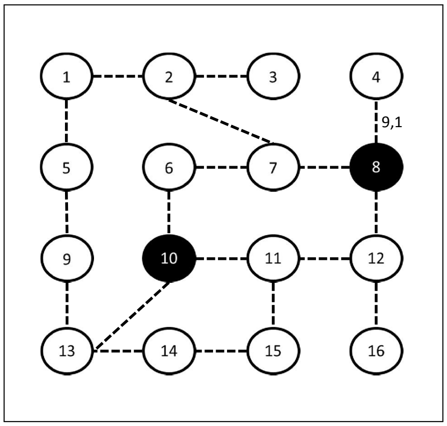

Figure A.19: Network state after incident and at the beginning of reconstruction. Solid lines represent functional links, while dashed lines represent damaged links that require reconstruction. All system parameters that are not shown are the same as in the base case.

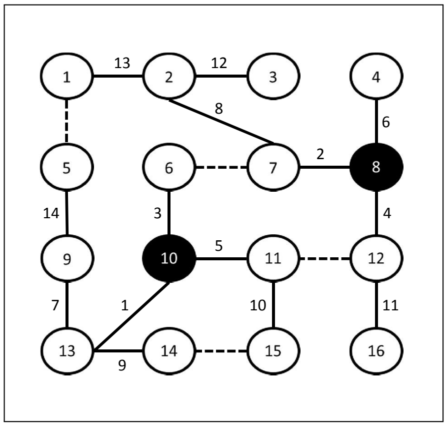

Figure A.20: Link labels indicate the optimal order of building the notional infrastructure system in Figure A.19.

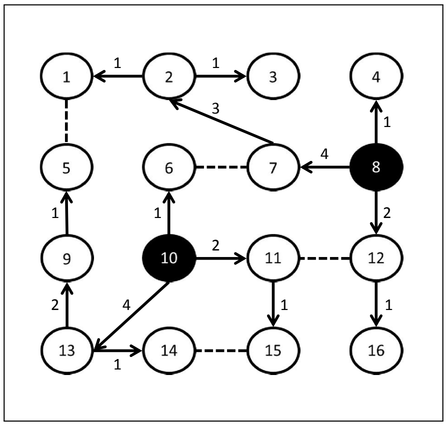

Figure A.21: Optimal distribution for fully operational notional infrastructure system in Figure A.19 with indicated direction and amount of flow for each link.

A.8 Utilize Unbroken Links to Build a Loop

At Figures A.22-A.24 we research again how to use operational links. Here we observe that the solution suggest to build a circle of operational links, which can be used to compensate broken links at later disasters.

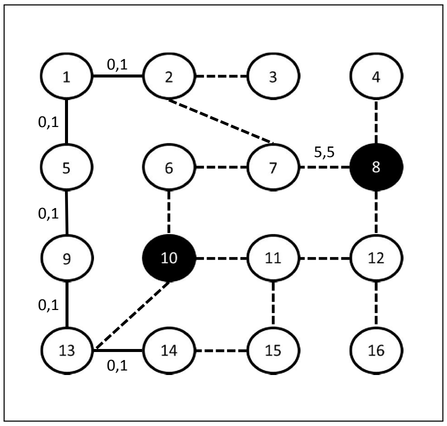

Figure A.22: Network state after incident and at the beginning of reconstruction. Solid lines represent functional links, while dashed lines represent damaged links that require reconstruction. All system parameters that are not shown are the same as in the base case.

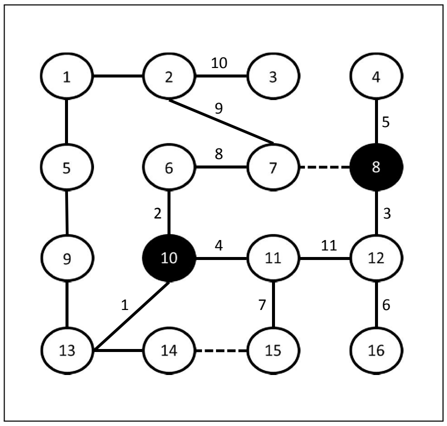

Figure A.23: Link labels indicate the optimal order of building the notional infrastructure system in Figure A.22.

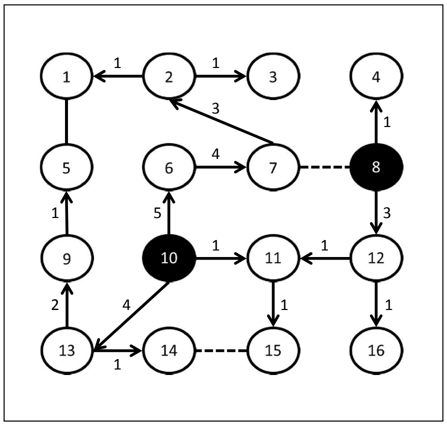

Figure A.24: Optimal distribution for fully operational notional infrastructure system in Figure A.22 with indicated direction and amount of flow for each link.

A.9 Not Using Unbroken Links for Transportation

At the experiment shown in Figures A.25-A.27 we see all links are broken except these which are not used for transportation at the base case. We observe that only link $[11,12]$ is used for transportation of commodities. This can be explained by the high building and transportation costs at link $[8,12]$.

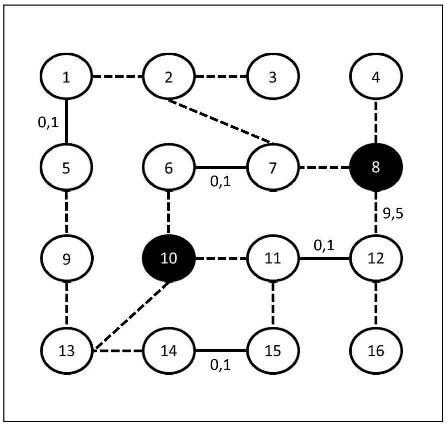

Figure A.25: Network state after incident and at the beginning of reconstruction. Solid lines represent functional links, while dashed lines represent damaged links that require reconstruction. All system parameters that are not shown are the same as in the base case.

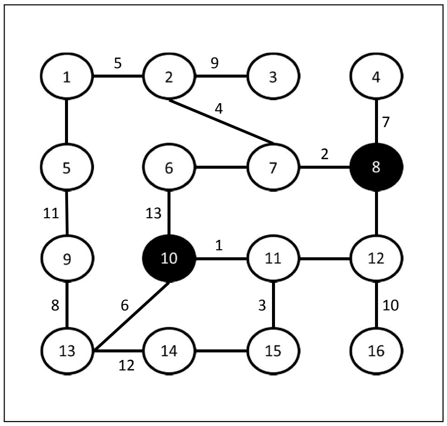

Figure A.26: Link labels indicate the optimal order of building the notional infrastructure system in Figure A.25.

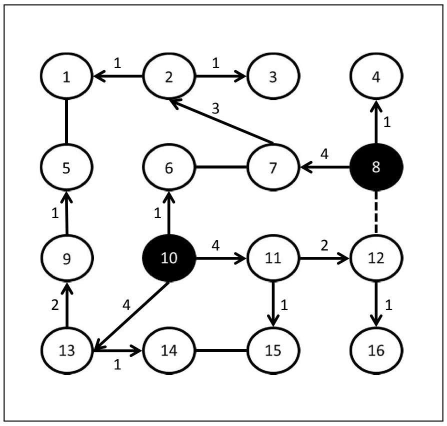

Figure A.27: Optimal distribution for fully operational notional infrastructure system in Figure A.25 with indicated direction and amount of flow for each link.

A.10 Not Building a Link because of High Transportation Cost

Figures A.28-A.30 show that if we increase the transportation cost at link $[6,10]$ this link will not be build. The demand at node 6 is fulfilled from the source node 8.

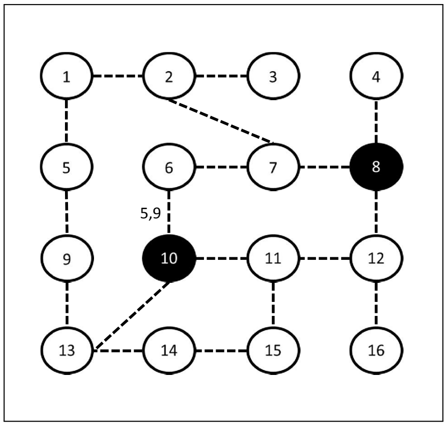

Figure A.28: Network state after incident and at the beginning of reconstruction. Solid lines represent functional links, while dashed lines represent damaged links that require reconstruction. All system parameters that are not shown are the same as in the base case.

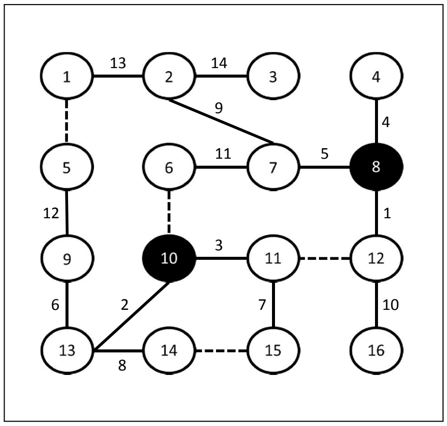

Figure A.29: Link labels indicate the optimal order of building the notional infrastructure system in Figure A.28.

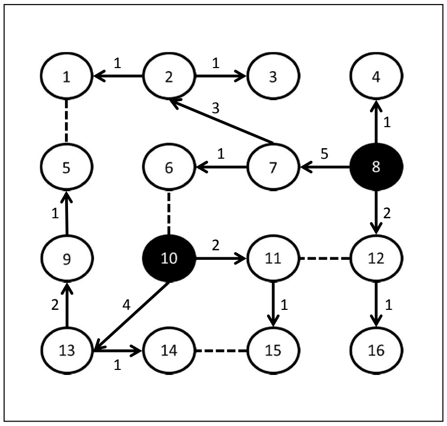

Figure A.30: Optimal distribution for fully operational notional infrastructure system in Figure A.28 with indicated direction and amount of flow for each link.

List of References

Alderson, D. L., G. G. Brown, W. M. Carlyle. 2014a. Assessing and improving operational resilience of critical infrastructures and other systems. A. Newman, J. Leung, eds., *Tutorials in Operations Research: Bridging Data and Decision*, vol. 745. Institute for Operations Research and Management Science, 180–215.

Alderson, D. L., G. G. Brown, W. M. Carlyle. 2014b. Operational models of infrastructure resilience. *Risk Analysis* **(to appear)**.

Álvarez, W. F., A. Vigo, H. J. Carlo, S. Long, T. Shoberg, S. Corns. 2014. A mathematical model for supply chain network infrastructure restoration. Y. Guan, H. Liao, eds., *Proceedings of the 2014 Industrial and Systems Engineering Research Conference*. Montréal (Québec) Canada. May 31 - June 1.

Baxter, M., T. Elgindy, A. T. Ernst, T. Kalinowski, M. WP. Savelsbergh. 2014. Incremental network design with shortest paths. *European Journal of Operational Research* **238**(3) 675–684.

Bigley, G. A., K. H. Roberts. 2001. The incident command system: High-reliability organizing for complex and volatile task environments. *Academy of Management Journal* **44**(6) 1281–1299.

Cavdaroglu, B., E. Hammel, J. E. Mitchell, T. C. Sharkey, W. A. Wallace. 2013. Integrating restoration and scheduling decisions for disrupted interdependent infrastructure systems. *Annals of Operations Research* **203**(1) 279–294.

Coffrin, C., P. Van Hentenryck, R. Bent. 2012. Last-mile restoration for multiple interdependent infrastructures. *Proceedings of the Association for the Advancement of Artificial Intelligence*. 455–463.

Davidson, R., Z. Cagnan. 2003. Post-earthquake lifeline service restoration modeling. *Advancing Mitigation Technologies and Disaster Response for Lifeline Systems*. ASCE, 255–264.

Dechter, R., J. Pearl. 1985. Generalized best-first search strategies and the optimality of A*. *Journal of the ACM (JACM)* **32**(3) 505–536.

Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *Numerische mathematik* **1**(1) 269–271.

Gallo, G., S. Pallottino. 1988. Shortest path algorithms. *Annals of Operations Research* **13**(1) 1–79.

GAMS Development Corp. 2014. GAMS v.24.0.1. Retrieved April 2013, from: www.gams.com.

Gong, J., E. E. Lee, J. E. Mitchell, W. A. Wallace. 2009. Logic-based multiobjective optimization for restoration planning. *Optimization and Logistics Challenges in the Enterprise*. Springer, 305–324.

Hart, P. E., N. J. Nilsson, B. Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on* **4**(2) 100–107.

Hart, W. E., C. Laird, J.-P. Watson, D. L. Woodruff. 2012. *Pyomo - Optimization Modeling in Python*. Springer Optimization and Its Applications, Vol. 67, Springer, New York.

Lee, E. E., J. E. Mitchell, W. A. Wallace. 2007. Restoration of services in interdependent infrastructure systems: A network flows approach. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **37**(6) 1303–1317.

Matisziw, T. C., A. T. Murray, T. H. Grubesic. 2010. Strategic network restoration. *Networks and Spatial Economics* **10**(3) 345–361.

NetworkX Developer Team. 2014. NetworkX: high-productivity software for complex networks. http://networkx.github.io.

Pederson, P., D. Dudenhoeffer, S. Hartley, M. Permann. 2006. Critical infrastructure interdependency modeling: a survey of us and international research. Tech. Rep. INL/EXT-0671464, Idaho National Laboratory.

Python Software Foundation. 2014. Python programming language. http://www.python.org.

Ran, Y. 2011. Considerations and suggestions on improvement of communication network disaster countermeasures after the Wenchuan earthquake. *Communications Magazine, IEEE* **49**(1) 44–47.

Reed, D. A., K. C. Kapur, R. D. Christie. 2009. Methodology for assessing the resilience of networked infrastructure. *Systems Journal, IEEE* **3**(2) 174–180.

Rinaldi, S. M. 2004. Modeling and simulating critical infrastructures and their interdependencies. *System sciences, 2004. Proceedings of the 37th annual Hawaii international conference on*. IEEE, 1–8.

Rinaldi, S. M., J. P. Peerenboom, T. K. Kelly. 2001. Identifying, understanding, and analyzing critical infrastructure interdependencies. *Control Systems, IEEE* **21**(6) 11–25.

Sanchez, A. 2010. Internet service provider network evolution in the presence of changing environmental conditions. Master's thesis, Naval Postgraduate School, Monterey, California.

Scanlon, T. J. 1994. The role of EOCs in emergency management: A comparison of Canadian and American experience. *International Journal of Mass Emergencies and Disasters* **12**(1) 51–75.

Shoji, G., A. Toyota. 2009. Modeling of restoration process associated with critical infrastructure and its interdependency due to a seismic disaster. *2009 Technical Council on Lifeline Earthquake Engineering: Lifeline Earthquake Engineering in a Multihazard Environment*. Oakland, CA, 647–658.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California